
QCVIEW Class Reference

[Cocoa > Graphics & Imaging](#)



2007-05-09



Apple Inc.
© 2007 Apple Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Apple, the Apple logo, Cocoa, Mac, Mac OS, and Quartz are trademarks of Apple Inc., registered in the United States and other countries.

OpenGL is a registered trademark of Silicon Graphics, Inc.

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

QCView Class Reference 7

Overview	7
Tasks	7
Performing Custom Operations During Rendering	7
Loading a Composition	8
Managing the Erase Color	8
Setting and Getting Event Masks	8
Setting and Getting the Maximum Frame Rate	8
Managing Rendering	8
Using Interface Builder	9
Taking Snapshot Images	9
Working With OpenGL	9
Instance Methods	9
autostartsRendering	9
createSnapshotImageOfType:	10
erase	10
eraseColor	11
eventForwardingMask	11
isPausedRendering	11
isRendering	12
loadComposition:	12
loadCompositionFromFile:	12
loadedComposition	13
maxRenderingFrameRate	13
openGLContext	14
openGLPixelFormat	14
pauseRendering	14
play:	15
renderAtTime:arguments:	15
resumeRendering	17
setAutostartsRendering:	17
setEraseColor:	18
setEventForwardingMask:	18
setMaxRenderingFrameRate:	19
snapshotImage	20
start:	20
startRendering	21
stop:	21
stopRendering	21
unloadComposition	22
Notifications	22

QCViewDidStartRenderingNotification 22
QCViewDidStopRenderingNotification 22

Document Revision History 23

Index 25

Tables

QCView Class Reference 7

Table 1 Events that can be forwarded to a composition 19

QCVIEW Class Reference

Inherits from	NSView : NSResponder : NSObject
Conforms to	QCCompositionRenderer NSAnimatablePropertyContainer (NSView) NSCoding (NSResponder) NSObject (NSObject)
Framework	/System/Library/Frameworks/Quartz.framework/Frameworks/QuartzComposer.framework
Availability	Available in Mac OS X v10.4 and later.
Companion guide	Quartz Composer Programming Guide
Declared in	QCVIEW.h

Overview

The `QCVIEW` class is a custom `NSVIEW` class that loads, plays, and controls Quartz Composer compositions. It is an autonomous view that is driven by an internal timer running on the main thread.

The view can be set to render a composition automatically when it is placed onscreen. The view stops rendering when it is placed offscreen. When not rendering, the view is filled with the current erase color. The rendered composition automatically synchronizes to the vertical retrace of the monitor.

When you archive a `QCVIEW` object, it saves the composition that's loaded at the time the view is archived.

If you want to perform custom operations while a composition is rendering such as setting input parameters or drawing OpenGL content, you need to subclass `QCVIEW` and implement the `renderAtTime:arguments:` (page 15) method.

Tasks

Performing Custom Operations During Rendering

- `renderAtTime:arguments:` (page 15)

Overrides to perform your custom operations prior to or after rendering a frame of a composition.

Loading a Composition

- [loadCompositionFromFile:](#) (page 12)
Loads the composition file located at the specified path.
- [loadComposition:](#) (page 12)
Loads a `QCComposition` object into the view.
- [loadedComposition](#) (page 13)
Returns the composition loaded in the view.
- [unloadComposition](#) (page 22)
Unloads the composition from the view.

Managing the Erase Color

- [erase](#) (page 10)
Clears the view using the current erase color.
- [eraseColor](#) (page 11)
Retrieves the current color used to erase the view.
- [setEraseColor:](#) (page 18)
Sets the color used to erase the view.

Setting and Getting Event Masks

- [eventForwardingMask](#) (page 11)
Retrieves the mask used to filter which types of events are forwarded from the view to the composition during rendering.
- [setEventForwardingMask:](#) (page 18)
Sets the mask used to filter which types of events are forwarded from the view to the composition during rendering.

Setting and Getting the Maximum Frame Rate

- [maxRenderingFrameRate](#) (page 13)
Returns the maximum frame rate for rendering.
- [setMaxRenderingFrameRate:](#) (page 19)
Sets the maximum rendering frame rate.

Managing Rendering

- [startRendering](#) (page 21)
Starts rendering the composition that is in the view.
- [isRendering](#) (page 12)
Checks whether a composition is rendering in the view.

- [autostartsRendering](#) (page 9)
Checks whether the view is set to start rendering automatically.
- [setAutostartsRendering:](#) (page 17)
Sets whether the composition that is in the view starts rendering automatically when the view is put on the screen.
- [stopRendering](#) (page 21)
Stops rendering the composition that is in the view.
- [pauseRendering](#) (page 14)
Pauses rendering in the view.
- [isPausedRendering](#) (page 11)
Returns whether or not the rendering in the view is paused.
- [resumeRendering](#) (page 17)
Resumes rendering a paused composition.

Using Interface Builder

- [play:](#) (page 15)
Plays or pauses a composition in a view.
- [start:](#) (page 20)
Starts rendering a composition in a view.
- [stop:](#) (page 21)
Stops rendering a composition in a view.

Taking Snapshot Images

- [snapshotImage](#) (page 20)
Returns an `UIImage` object of the current image in the view.
- [createSnapshotImageOfType:](#) (page 10)
Returns the current image in the view as an image object of the provided image type.

Working With OpenGL

- [openGLContext](#) (page 14)
Returns the OpenGL context used by the view.
- [openGLPixelFormat](#) (page 14)
Returns the OpenGL pixel format used by the view.

Instance Methods

autostartsRendering

Checks whether the view is set to start rendering automatically.

- (BOOL)autostartsRendering

Return Value

Returns YES if the view is set to start rendering automatically when the view is put on screen.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setAutostartsRendering:](#) (page 17)

Declared In

QCVIEW.h

createSnapshotImageOfType:

Returns the current image in the view as an image object of the provided image type.

- (id) createSnapshotImageOfType:(NSString*)type

Parameters

type

A string that specifies any of the following image types: NSBitmapImageRep, NSImage, CIImage, CGImage, CVOpenGLBuffer, CVPixelBuffer.

Return Value

The snapshot image in the provided image type. You are responsible for releasing this object when you no longer need it.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [snapshotImage](#) (page 20)

Declared In

QCVIEW.h

erase

Clears the view using the current erase color.

- (void)erase

Availability

Available in Mac OS X v10.4 and later.

See Also

- [eraseColor](#) (page 11)

Declared In

QCVIEW.h

eraseColor

Retrieves the current color used to erase the view.

- (NSColor *)eraseColor

Return Value

The color object previously set using the [setEraseColor:](#) (page 18) method.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [erase](#) (page 10)

Declared In

QCView.h

eventForwardingMask

Retrieves the mask used to filter which types of events are forwarded from the view to the composition during rendering.

- (NSUInteger)eventForwardingMask

Return Value

The event filtering mask.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setEventForwardingMask:](#) (page 18)

Declared In

QCView.h

isPausedRendering

Returns whether or not the rendering in the view is paused.

- (BOOL) isPausedRendering;

Return Value

YES if the rendering is paused; otherwise NO.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [pauseRendering](#) (page 14)

- [resumeRendering](#) (page 17)

Declared In

QCView.h

isRendering

Checks whether a composition is rendering in the view.

- (BOOL)isRendering

Return Value

Returns YES if a composition is rendering in the view; NO otherwise.

Availability

Available in Mac OS X v10.5 and later.

Declared In

QCView.h

loadComposition:Loads a `QCComposition` object into the view.

- (BOOL)loadComposition:(QCComposition*)composition

Parameters*composition*The `QCComposition` object to load.**Return Value**

YES if successful; otherwise NO. If unsuccessful, any composition that's already loaded in the view remains loaded.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [loadCompositionFromFile:](#) (page 12)
- [unloadComposition](#) (page 22)
- [loadedComposition](#) (page 13)

Declared In

QCView.h

loadCompositionFromFile:

Loads the composition file located at the specified path.

- (BOOL)loadCompositionFromFile:(NSString *)path

Parameters*path*

A string that specifies the location of a Quartz Composer composition file.

Return Value

If unsuccessful, returns NO; any composition that's already loaded in the view remains loaded.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [loadComposition:](#) (page 12)
- [unloadComposition](#) (page 22)
- [loadedComposition](#) (page 13)

Declared In

QCVIEW.h

loadedComposition

Returns the composition loaded in the view.

- (QCCOMPOSITION*) loadedComposition

Return Value

The composition loaded in the view; otherwise nil.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [loadCompositionFromFile:](#) (page 12)
- [loadComposition:](#) (page 12)
- [unloadComposition](#) (page 22)

Declared In

QCVIEW.h

maxRenderingFrameRate

Returns the maximum frame rate for rendering.

- (float)maxRenderingFrameRate

Return Value

The maximum frame rate for rendering. A value of 0.0 specifies that there is no limit.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setMaxRenderingFrameRate:](#) (page 19)

Declared In

QCVIEW.h

openGLContext

Returns the OpenGL context used by the view.

- (NSOpenGLContext*) openGLContext

Return Value

An `NSOpenGLContext` object.

Discussion

This context as a read-only object . Do not attempt to change any of its settings. If you subclass `QCView` so that you can perform custom OpenGL drawing, you'll need to use this method to retrieve the view's OpenGL context.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [renderAtTime:arguments:](#) (page 15)

Declared In

`QCView.h`

openGLPixelFormat

Returns the OpenGL pixel format used by the view.

- (NSOpenGLPixelFormat*) openGLPixelFormat

Return Value

An `NSOpenGLPixelFormat` object.

Discussion

This pixel format as a read-only object. Do not attempt to change any of its settings.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`QCView.h`

pauseRendering

Pauses rendering in the view.

- (void) pauseRendering

Discussion

You can nest calls to this method.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [resumeRendering](#) (page 17)
- [isPausedRendering](#) (page 11)

Declared In

QCView.h

play:

Plays or pauses a composition in a view.

- (IBAction) play:(id)sender

Parameters*sender*

The object (such as a button or menu item) sending the message to play the composition. You need to connect the object in the interface to the action.

Return Value

The message sent to the target.

Discussion

This method starts rendering a composition if it is not already rendering, pauses a composition that is rendering, or resumes rendering for a composition whose rendering is paused. The method is invoked when the user clicks a button or issues a command from some other user interface element, such as a menu.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [stop:](#) (page 21)

Declared In

QCView.h

renderAtTime:arguments:

Overrides to perform your custom operations prior to or after rendering a frame of a composition.

- (BOOL) renderAtTime:(NSTimeInterval)time arguments:(NSDictionary*)arguments

Parameters*time*

The rendering time, in seconds, of the composition frame.

arguments

An optional dictionary that can contain `QCRendererEventKey` or `QCRendererMouseLocationKey` and the associated values. (See *QCRenderer Class Reference* or more information.)

Return Value

NO if your custom rendering fails, otherwise, YES.

Discussion

Do not call this method directly. You override this method only for subclasses of the `QCView` class and only if you want to perform custom operations or OpenGL rendering before and/or after Quartz Composer renders a frame of the composition.

The most common reasons to override this method are to:

- synchronize communication with the composition. For example, you might want to set input parameters of the composition. By overriding this method, you can set parameters only when necessary and only at a specific time.
- underlay or overlay custom OpenGL rendering.

To synchronize communication between a composition and another part of the application, the implementation looks similar to the following:

```
- (BOOL) renderAtTime:(NSTimeInterval)time
    arguments:(NSDictionary*)arguments
{
    // Your code to compute the value of myParameterValue
    [self setValue:myParameterValue forKey:@"myInput"];

    BOOL success = [super renderAtTime:time arguments:arguments];

    id result = [self valueForKey:@"myOutput"];
    //Your code to perform some operation on the result

    return success;
}
```

To perform OpenGL drawing in a `QCView` object, follow these guidelines:

- Use the OpenGL context of the `QCView` object to do drawing. You can retrieve the OpenGL context by calling `[self openGLContext]`. Note that this context won't necessarily be set as the current OpenGL context.
- Use CGL macros instead of managing the current OpenGL context yourself.

OpenGL performs a global context and renderer lookup for each command it executes to ensure that all OpenGL commands are issued to the correct rendering context and renderer. There is significant overhead associated with these lookups that can measurably affect performance. CGL macros let you provide a local context variable and cache the current renderer in that variable. They are simple to use, taking only a few lines of code to set up.
- Save and restore all state changes except the ones that are part of `GL_CURRENT_BIT` (RGBA color, color index, normal vector, texture coordinates, and so forth).
- Check for OpenGL errors with `glGetError`.

Here's an example implementation of this method using OpenGL to draw an overlay:

```
#import <OpenGL/CGLMacro.h> // Set up using macros

- (BOOL) renderAtTime:(NSTimeInterval)time
    arguments:(NSDictionary*)arguments
{
```



```

    BOOL success = [super renderAtTime:time arguments:arguments];

    // Use the OpenGL context of the view for drawing.
    CGLContextObj cgl_ctx = [[self openGLContext] CGLContextObj];

    // Save and set OpenGL states appropriately.
    glGetIntegerv(GL_MATRIX_MODE, &saveMode);
    glMatrixMode(GL_MODELVIEW);
    glPushMatrix();
    glRotatef(45.0, 0.0, 0.0, 1.0);

    // The code that performs OpenGL drawing goes here.
    //After drawing, restore original OpenGL states.
    glPopMatrix();
    glMatrixMode(saveMode);

    // Check for errors.
    glGetError();
    return success;
}

```

Availability

Available in Mac OS X v10.5 and later.

Declared In

QCView.h

resumeRendering

Resumes rendering a paused composition.

- (void) resumeRendering

Discussion

You can nest calls to this method.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [pauseRendering](#) (page 14)
- [isPausedRendering](#) (page 11)

Declared In

QCView.h

setAutostartsRendering:

Sets whether the composition that is in the view starts rendering automatically when the view is put on the screen.

- (void) setAutostartsRendering:(BOOL) flag

Parameters

flag

Pass YES to enable autostart mode; NO otherwise.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [autostartsRendering](#) (page 9)

Declared In

QCView.h

setEraseColor:

Sets the color used to erase the view.

- (void)setEraseColor:(NSColor *)*color*

Parameters

color

A color object.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [erase](#) (page 10)

- [eraseColor](#) (page 11)

Declared In

QCView.h

setEventForwardingMask:

Sets the mask used to filter which types of events are forwarded from the view to the composition during rendering.

- (void)setEventForwardingMask:(NSUInteger)*mask*

Parameters

mask

An event filtering mask. The mask can be a combination of any of the mask constants listed in Table 1 or the constant `NSAnyEventMask`.

Table 1 Events that can be forwarded to a composition

Event	Description
<code>NSLeftMouseDownMask</code>	The user pressed the left button.
<code>NSLeftMouseDraggedMask</code>	The user moved the mouse with the left button down.
<code>NSLeftMouseUpMask</code>	The user released the left button.
<code>NSRightMouseDownMask</code>	The user pressed the right button.
<code>NSRightMouseDraggedMask</code>	The user moved the mouse with the right button down.
<code>NSRightMouseUpMask</code>	The user released the right button.
<code>NSOtherMouseDownMask</code>	The user pressed the middle button, or some button other than the left or right button.
<code>NSOtherMouseDraggedMask</code>	The user moved the mouse with the middle button down, or some button other than the left or right button.
<code>NSOtherMouseUpMask</code>	The user released the middle button, or some button other than the left or right button.
<code>NSMouseMovedMask</code>	The user moved the mouse without holding down a mouse button.
<code>NSScrollWheelMask</code>	The user moved the mouse scroll wheel.
<code>NSKeyDownMask</code>	The user generated a character or characters by pressing a key.
<code>NSKeyUpMask</code>	The user released a key.
<code>NSFlagsChangedMask</code>	The user pressed or released a modifier key, or toggled the Caps Lock key.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [eventForwardingMask](#) (page 11)

Declared In

QCView.h

setMaxRenderingFrameRate:

Sets the maximum rendering frame rate.

- (void)setMaxRenderingFrameRate:(float)maxFPS

Parameters

maxFPS

The frame rate to set. Pass 0.0 to specify that there is no limit.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [maxRenderingFrameRate](#) (page 13)

Declared In

QCVIEW.h

snapshotImage

Returns an NSImage object of the current image in the view.

- (NSImage*) snapshotImage

Return Value

The snapshot image.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [createSnapshotImageOfType:](#) (page 10)

Declared In

QCVIEW.h

start:

Starts rendering a composition in a view.

- (IBAction)start:(id) sender

Parameters

sender

The object (such as a button or menu item) sending the message to start rendering. You need to connect the object in the interface to the action.

Return Value

The message sent to the target.

Discussion

The method is invoked when the user clicks a button or issues a command from some other user interface element, such as a menu. It is equivalent to the [startRendering](#) (page 21) method.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [stop:](#) (page 21)

Declared In

QCView.h

startRendering

Starts rendering the composition that is in the view.

- (BOOL)startRendering

Return Value

Returns NO if the composition fails to start rendering; YES otherwise.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [stopRendering](#) (page 21)

Declared In

QCView.h

stop:

Stops rendering a composition in a view.

- (IBAction)stop:(id)sender

Parameters

sender

The object (such as a button or menu item) sending the message to stop rendering. You need to connect the object in the interface to the action.

Return Value

The message sent to the target.

Discussion

The method is invoked when the user clicks a button or issues a command from some other user interface element, such as a menu. It is equivalent to the [stopRendering](#) (page 21) method.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [start:](#) (page 20)

Declared In

QCView.h

stopRendering

Stops rendering the composition that is in the view.

- (void)stopRendering

Availability

Available in Mac OS X v10.4 and later.

See Also

- [startRendering](#) (page 21)

Declared In

QCVIEW.h

unloadComposition

Unloads the composition from the view.

- (void) unloadComposition;

Discussion

If necessary, this method calls [stopRendering](#) (page 21) prior to unloading the composition.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [loadCompositionFromFile:](#) (page 12)

- [loadComposition:](#) (page 12)

- [loadedComposition](#) (page 13)

Declared In

QCVIEW.h

Notifications

QCVIEWDidStartRenderingNotification

Posted when the view starts rendering.

Availability

Available in Mac OS X v10.4 and later.

Declared In

QCVIEW.h

QCVIEWDidStopRenderingNotification

Posted when the view stops rendering.

Availability

Available in Mac OS X v10.4 and later.

Declared In

QCVIEW.h

Document Revision History

This table describes the changes to *QCVIEW Class Reference*.

Date	Notes
2007-05-09	Added links and improved class metadata.
	Added loadComposition : (page 12), loadedComposition (page 13), unloadComposition (page 22), renderAtTime:arguments: (page 15), pauseRendering (page 14), isPausedRendering (page 11), resumeRendering (page 17), snapshotImage (page 20), createSnapshotImageOfType: (page 10), openGLContext (page 14), openGLPixelFormat (page 14), and play: (page 15).
	Removed <code>setValue:forInputKey:</code> , <code>valueForInputKey:</code> , and <code>valueForOutputKey:</code> . Instead see <i>QCRenderer Class Reference</i> .
2006-05-23	First publication of this content as a separate document.
	Edited content to comply with new guidelines. Added See Also sections and parameter descriptions.

REVISION HISTORY

Document Revision History

Index

A

autostartsRendering [instance method 9](#)

C

createSnapshotImageOfType: [instance method 10](#)

E

erase [instance method 10](#)
eraseColor [instance method 11](#)
eventForwardingMask [instance method 11](#)

I

isPausedRendering [instance method 11](#)
isRendering [instance method 12](#)

L

loadComposition: [instance method 12](#)
loadCompositionFromFile: [instance method 12](#)
loadedComposition [instance method 13](#)

M

maxRenderingFrameRate [instance method 13](#)

O

openGLContext [instance method 14](#)

openGLPixelFormat [instance method 14](#)

P

pauseRendering [instance method 14](#)
play: [instance method 15](#)

Q

QCVIEWDIDSTARTRENDERINGNOTIFICATION [notification 22](#)
QCVIEWDIDSTOPRENDERINGNOTIFICATION [notification 22](#)

R

renderAtTime:arguments: [instance method 15](#)
resumeRendering [instance method 17](#)

S

setAutostartsRendering: [instance method 17](#)
setEraseColor: [instance method 18](#)
setEventForwardingMask: [instance method 18](#)
setMaxRenderingFrameRate: [instance method 19](#)
snapshotImage [instance method 20](#)
start: [instance method 20](#)
startRendering [instance method 21](#)
stop: [instance method 21](#)
stopRendering [instance method 21](#)

U

unloadComposition [instance method 22](#)